

# Visual Context Awareness via Wearable Computing

Thad Starner, Bernt Schiele, and Alex Pentland  
Media Laboratory, Massachusetts Institute of Technology  
{testarne,bernt,sandy}@media.mit.edu

## Abstract

*Small, body-mounted video cameras enable a different style of computing interface. As processing power increases, a computer can spend more time observing its user and modeling his or her context to provide serendipitous information, manage interruptions and tasks, and predict future needs without being directly commanded by the user. This paper introduces a wearable assistant, driven by computer vision, for the real-space game Patrol. The goal of the system is to be aware of the user context by tracking the wearer's location and current task through computer vision techniques.*

## 1. Introduction

For most computer systems, even virtual reality systems, sensing techniques are a means of getting input directly from the user. However, wearable sensors and computers offer a unique opportunity to re-direct sensing technology towards recovering more general user context. Wearable computers have the potential to “see” as the user sees, “hear” as the user hears, and experience the life of the user in a “first-person” sense. This increase in contextual and user information may lead to more intelligent and fluid interfaces that use the physical world as part of the interface.

The importance of context in communication and interface can not be overstated. Physical environment, time of day, mental state, and the model each conversant has of the other participants can be critical in conveying necessary information and mood. Imagine an interface which is aware of where you are: While being in the subway, the system might alert the user with a spoken summary of an e-mail. However, during a conversation, the wearable computer may present the name of a potential caller unobtrusively in the user's head-up display. The more understanding the computer has about the context of the user the better the computer may interact with the human.

In this paper we propose to use wearable cameras to sense and model user context. The on-body cameras not only have a “first person”-view but also enable observation of the user's actions and tasks. In this sense the paper shows how computer interfaces may become more contextually aware through machine vision techniques. Section 2 describes the importance of object identification in combining the virtual environ-

ment of the computer with the physical environment of the user. Section 3.1 details how the location of the user may provide salient cues to his current context for the real-space game Patrol. Finally, a means of determining the user's current task in Patrol is discussed in Section 3.2. See [12] for a more thorough treatment of this topic.

## 2. Identification of Relevant Objects

One of the most distinctive advantages of wearable computing is the coupling of the virtual environment with the physical world. Thus, determining the presence and location of physical objects relative to the user is an important task. Once an object is uniquely labeled, the user's wearable computer can assign virtual properties to the object, such as annotations or hypertext links [11].

One method for object identification is Radio Frequency Identification (RFID transmitter tag with a unique ID attached to each object to be tracked). Unfortunately, this method requires a significant amount of physical infrastructure and maintenance for placing and reading the tags.

Computer vision not only obviates the need for expensive tags but also adapts to different scales and ranges. For example, the same hardware and software may recognize a thimble or a building depending on the distance of the camera to the object. In addition, computer vision is directed. By aligning the field of view of the camera with the field of view of the eye, the computer may observe the objects that are the focus of the user's attention.

In the past, the MIT Wearable Computing Project has used computer vision identification to create a physically-based hypertext demonstration platform [11] based on colored tags. [4] describes a similar identification system for a tethered, hand-held system. The DyPERS system [8] demonstrates how visual tags become unnecessary when a more sophisticated object recognition system is employed.

Such physically-based annotation systems are appropriate for tours of a city or a museum. As reliability and accessibility to wireless networks improves, such systems might be used for repair, inspection, and maintenance of hidden physical infrastructure, such as electrical wiring or plumbing [1, 11]. Similarly, AR systems might be used as navigation guides or task reminders for the mentally handicapped. As recognition performance increases and the hardware costs

decline, many new applications will be found for such contextually-aware computing.

### 3. The Patrol Task

The “Patrol task” is an attempt to test techniques from the laboratory in less constrained environments. Patrol is a game played by MIT students every weekend in a campus building. The participants are divided into teams denoted by colored head bands. Each participant starts with a rubber suction dart gun and a small number of darts. The goal is to hunt the other teams. If shot with a dart, the participant removes his head band, waits for fighting to finish, and proceeds to the second floor before replacing his head band and returning.

Originally, Patrol provided an entertaining way to test the robustness of wearable computing techniques and apparatus for other projects, such as hand tracking for the sign language recognizer [13]. However, it quickly became apparent that the gestures and actions in Patrol provided a relatively well defined language and goal structure in a very harsh “real-life” sensing environment. As such, Patrol became a context-sensing project within itself. The next sections discuss current work on determining player location and task using only on-body sensing apparatus.

Sensing for the Patrol task is performed by two hat-mounted wide-angle cameras (Figure 1). The larger of the two cameras points downwards to watch the hands and body. The smaller points forward to observe what the user sees. Figure 1 shows sample images from the hat. While it is possible to provide enough on-body computation to run feature detection in real-time, we currently record to video tape during the game for experimental purposes.

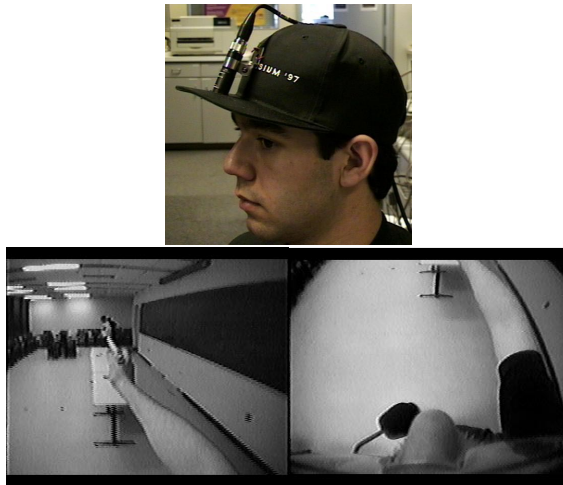
#### 3.1. Location

As mentioned earlier, user location often provides valuable clues to the user’s context. By gathering data over many days, the user’s motions throughout the day might be modeled. This model may then be used to predict when the user will be in a certain location and for how long [5].

Today, most outdoor positioning is performed in relation to the Global Positioning System (GPS). Differential systems can obtain accuracies on the order of centimeters. Current indoor systems such as active badges [14, 2] and beacon architectures [3, 9, 10] require increased infrastructure for higher accuracy, implying increased installation and maintenance. Here, we attempt to determine location based solely on the images provided by the Patrol hat cameras, which are fixed-cost on-body equipment.

The Patrol environment consists of 14 rooms that are defined by their strategic importance to the players. The rooms’ boundaries were not chosen to simplify the vision task but are based on the long standing conventions of game play. The playing areas include hallways, stairwells, classrooms, and mirror image copies of these

classrooms whose similarities and “institutional” decor make the recognition task difficult.



**Figure 1. Above: The two camera Patrol hat. Below: the downward- and forward-looking Patrol views.**

Hidden Markov models (HMM’s) were chosen to represent the environment due to their potential language structure and excellent discrimination ability for varying time domain processes. For example, rooms may have distinct regions or lighting that can be modeled by the states in an HMM. In addition, the previous known location of the user helps to limit his current possible location. By observing the video stream over several minutes and knowing the physical layout of the building and the mean time spent in each area, many possible paths may be hypothesized and the most probable chosen based on the observed data. HMM’s fully exploit these attributes. For a review of HMM’s see [6].

As a first attempt, the means of the red, green, blue, and luminance pixel values of three image patches are used to construct a feature vector in real-time. One patch is taken from the image of the forward looking camera. This patch varies significantly due to the head motion of the player. The next patch represents the coloration of the floors and is derived from the downward looking camera in the area just to the front of the player and out of range of average hand and foot motion. Finally, since the nose is always in the same place relative to the downward looking camera, a patch is sampled from the nose, providing information about the lighting variations as the player moves through a room.

Approximately 45 minutes of annotated Patrol video were analyzed for this experiment (six frames per second). 24.5 minutes of video, comprising 87 area transitions, are used for training the HMMs. As part of the training, a statistical (bigram) grammar is generated. This “grammar” is used in testing to weight those rooms which are considered next based on the current hypothesized room. An independent 19.3 min-

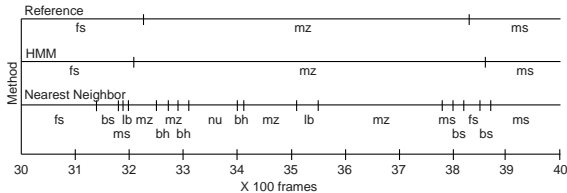
utes of video, comprising 55 area transitions, are used for testing. Note that the computer must segment the video at the area transitions as well as label the areas properly.

Table 1 demonstrates the accuracies of the different methods tested. For informative purposes, accuracy rates are reported both for testing on the training data and the independent test set. Accuracy is calculated by  $Acc = \frac{N-D-S-I}{N}$ , where  $N$  is the total number of areas in the test set,  $D$  (deletions) is the number of area changes not detected,  $S$  (substitutions) is the number of areas falsely labeled, and  $I$  (insertions) is the number of area transitions falsely detected. Note that, since all errors are counted against the accuracy rate, it is possible to get large negative accuracies by having many insertions.

**Table 1. Patrol area recognition accuracy**

<i>method</i>	<i>training set</i>	<i>test set</i>
2-state HMM	51.72%	21.82%
3-state HMM	68.97%	81.82%
4-state HMM	65.52%	76.36%
5-state HMM	79.31%	40.00%
Nearest Neighbor	-400%	-485.18%

The simplest method for determining the current room is to determine the smallest Euclidean distance between a test feature vector with the means of the feature vectors comprising the different room examples in the training set. Given this nearest neighbor method as a comparison, it is easy to see how the time duration and contextual properties of the HMM’s improve recognition. Testing on the independent test set shows that the best model is a 3-state HMM, which achieves 82% accuracy. In some cases accuracy on the test data is better than the training data, probably due to changing video quality from falling battery voltage.



**Figure 2. Typical detection of Patrol area transitions.**

Another important attribute is how well the system determines when the player has entered a new area. Figure 2 compares the 3-state HMM and nearest neighbor methods to the hand-labeled video. Different rooms are designated by two letter identifiers. As can be seen, the 3-state HMM system tends to be within a few seconds of the correct transition boundaries while the nearest neighbor system oscillates between many hypotheses.

As mentioned earlier, one of the strengths of the HMM system is that it can collect evidence over time to hypothesize the player’s path through several areas. How much difference does this incorporation of context make on recognition? To determine this, the test set was segmented by hand, and each area was presented in isolation to the 3-state HMM system. At face value this should be a much easier task since the system does not have to segment the areas as well as recognize them. However, the system only achieved 49% accuracy on the test data and 78% accuracy on the training data. This result provides striking evidence of the importance of using context in this task and hints at the importance of context in other user activities.

### 3.2. User Tasks

By identifying the user’s current task, the computer can assist actively in that task by displaying timely information or automatically reserving resources that may be needed [1, 11]. However, a wearable computer might also take a more passive role, simply determining the importance of potential interruptions (phone, e-mail, paging, etc.) and presenting the interruption in the most socially graceful manner possible.

Here we describe an experiment to recognize the user tasks aiming, reloading and “other” tasks. In order to recognize such user tasks we use a generic object recognition system recently proposed by Schiele and Crowley (see [7] for details). In the context of the Patrol data this system can be used for recognition of image patches that correspond to particular motions of a hand, the gun, a portion of an arm, or any part of the background. By feeding the calculated probabilities as feature vectors to a set of hidden Markov models (HMM’s), it is possible to recognize different user tasks such as aiming and reloading.

In order to use the recognition system we define a library of images grouped into images corresponding to the same action. Each image is split into 4x4 sub-images used as image patch database. In the experiment below we define three different image groups, one of each action so that the system calculates 3 groups×16 = 48 probabilities at 10Hz. These probabilities are then used as feature vector for a set of HMM’s trained to recognize different tasks of the user.

For two actions (aiming and reloading) we train a separate HMM containing 5 states on an annotated 2 minutes video segment containing 13 aiming actions and 6 reloading actions. Everything which is neither aiming nor reloading is modeled by a third class, the “other” class (10 sequences in total). The actions have been separated into a training set of 7 aiming actions, 4 reloading actions and 3 other sequences for training of the HMM’s. Interestingly, the actions are of very different length (between 2.25sec and 0.3sec). The remaining actions have been used as test set. Table 2 shows the confusion matrix of the three action classes.

Aiming is relatively distinctive with respect to reloading and “other”, since the arm is stretched out during aiming, which is probably the reason for the

**Table 2. Confusion matrix between aiming, reloading, and other tasks.**

	aiming	reloading	“other”
aiming	6	0	0
reloading	0	1	1
“other”	0	1	6

perfect recognition of the aiming sequences. However, reloading and “other” are difficult to distinguish, since the reloading action happens only in a very small region of the image (close to the body) and is sometimes barely visible.

These preliminary results are certainly encouraging, but have been obtained for perfectly segmented data and a very small set of actions. However, an intrinsic property of HMM’s is that they generalize to unsegmented data well. Furthermore the increase of the task vocabulary will enable the use of language and context models which will help the recognition of single tasks.

### 3.3. Use of Patrol Context

While preliminary, the systems described above suggest interesting interfaces. By using head-up displays, the players could keep track of each other’s locations. A strategist can deploy the team as appropriate for maintaining territory. If aim and reload gestures are recognized for a particular player, the computer can automatically alert nearby team members for aid.

Contextual information can be used more subtly as well. For example, if the computer recognizes that its wearer is in the middle of a skirmish, it should inhibit all interruptions and information. Similarly, a simple optical flow algorithm may be used to determine when the player is scouting a new area. Again, any interruption should be inhibited. On the other hand, when the user is “resurrecting” or waiting, the computer should provide as much information as possible to prepare the user for rejoining the game.

The model created by the HMM location system above can also be used for prediction. For example, the computer can weight the importance of incoming information depending on where it believes the player will move next. An encounter among other players several rooms away may be relevant if the player is moving rapidly in that direction. In addition, if the player is shot, the computer may predict the most likely next area for the enemy to visit and alert the player’s team as appropriate. Such just-in-time information can be invaluable in such hectic situations.

## 4. Conclusion and Future Work

Through body centered cameras and machine vision techniques, several examples of contextually aware interfaces are presented. By observing context, the computer can aid in task and interruption management, provide just-in-time information, and make helpful predictions of future behavior. While larger annotated

data sets are necessary to test the techniques used for the Patrol task, the preliminary results are promising. Additional methods such as optical flow or motion differencing will be added to determine if the user is standing, walking, running, visually scanning the scene, or using the stairs. By using the new apparatus to analyze video and audio from two simultaneous participants, player interaction might be modeled. Hopefully, with development, such a system will be used to observe and model everyday user tasks and human to human interactions as well.

## References

- [1] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, 1993.
- [2] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In *FRIEND21: Inter. Symp. on Next Generation Human Interface*, pages 125–128, 1994.
- [3] S. Long, R. Kooper, G. Abowd, and C. Atkeson. Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *MobiCom*. ACM Press, 1996.
- [4] K. Nagao and J. Rekimoto. Ubiquitous talker: Spoken language interaction with real world objects. In *IJCAI*, pages 1284–1290, Montreal, 1995.
- [5] J. Orwant. Doppelganger goes to school: Machine learning for user modeling. Master’s thesis, MIT, Media Laboratory, September 1993.
- [6] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [7] B. Schiele and J. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *International Conf. on Pat. Rec.*, volume B, pages 50–54, August 1996.
- [8] B. Schiele, N. Oliver, T. Jebara, and A. Pentland. An interactive computer vision system, dypers: Dynamic personal enhanced remembrance system. In *International Conference on Vision Systems*, Jan 1999.
- [9] W. Schilit. *System architecture for context-aware mobile computing*. PhD thesis, Columbia University, 1995.
- [10] T. Starner, D. Kirsch, and S. Assefa. The locust swarm: An environmentally-powered, networkless location and messaging system. In *ISWC*, 1997.
- [11] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence*, 6(4):386–398, Winter 1997.
- [12] T. Starner, B. Schiele, and A. Pentland. Visual context awareness in wearable computing. Technical Report 465, MIT MediaLab, Vismod group, 1998.
- [13] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer-based video. *PAMI*, To appear 1998.
- [14] R. Want and A. Hopper. Active badges and personal interactive computing objects. *IEEE Trans. on Consumer Electronics*, 38(1):10–20, Feb. 1992.